



# Multivariada II

---

- *Profa. MSc. Edmila Montezani*
- *edmila@gmail.com*



# O que são variáveis categóricas, discretas e contínuas?

---

Variáveis quantitativas podem ser classificadas como discretas ou contínuas.

## **Variável categórica**

As variáveis categóricas contêm um número finito de categorias ou grupos distintos. Os dados categóricos podem não ter uma ordem lógica. Por exemplo, os preditores categóricos incluem gênero, tipo de material e método de pagamento.

## **Variável discreta**

Variáveis discretas são variáveis numéricas que têm um número contável de valores entre quaisquer dois valores. Uma variável discreta é sempre numérica. Por exemplo, o número de reclamações de clientes ou o número de falhas ou defeitos.

## **Variável contínua**

Variáveis contínuas são variáveis numéricas que têm um número infinito de valores entre dois valores quaisquer. Uma variável contínua pode ser numérica ou de data/hora. Por exemplo, o comprimento de uma peça ou a data e hora em que um pagamento é recebido.



## Exemplo 1: infert

A primeira base de dados que vamos utilizar são os dados de um estudo caso-controle, em que os casos foram mulheres com infertilidade e os controles, mulheres sem infertilidade. Os principais fatores de risco a serem analisados são os abortos naturais e induzidos. As variáveis idade, escolaridade e paridade são consideradas de controle.

- **Base de dados de exemplo**

```
>library(help=datasets)
```

- **Importar dados**

```
>data(infert)
```

```
> infert
```

	education	age	parity	induced	case	spontaneous	stratum	pooled.stratum
1	0-5yrs	26	6	1	1	2	1	3
2	0-5yrs	42	1	1	1	0	2	1
3	0-5yrs	39	6	2	1	0	3	4
4	0-5yrs	34	4	2	1	0	4	2
5	6-11yrs	35	3	1	1	1	5	32
6	6-11yrs	36	4	2	1	1	6	36
7	6-11yrs	23	1	0	1	0	7	6
8	6-11yrs	32	2	0	1	0	8	22
9	6-11yrs	21	1	0	1	1	9	5
10	6-11yrs	28	2	0	1	0	10	19
11	6-11yrs	20	2	1	1	0	11	20

Environment

History

Connections

Import Dataset

Global Environment

Data

infert	248 obs. of 8 variables
--------	-------------------------

# Distribuição de frequência das variáveis em infert

```
> summary(infert)
```

education	age	parity	induced	case	spontaneous
0-5yrs : 12	Min. :21.00	Min. :1.000	Min. :0.0000	Min. :0.0000	Min. :0.0000
6-11yrs:120	1st Qu.:28.00	1st Qu.:1.000	1st Qu.:0.0000	1st Qu.:0.0000	1st Qu.:0.0000
12+ yrs:116	Median :31.00	Median :2.000	Median :0.0000	Median :0.0000	Median :0.0000
	Mean :31.50	Mean :2.093	Mean :0.5726	Mean :0.3347	Mean :0.5766
	3rd Qu.:35.25	3rd Qu.:3.000	3rd Qu.:1.0000	3rd Qu.:1.0000	3rd Qu.:1.0000
	Max. :44.00	Max. :6.000	Max. :2.0000	Max. :1.0000	Max. :2.0000

stratum	pooled.stratum
Min. : 1.00	Min. : 1.00
1st Qu.:21.00	1st Qu.:19.00
Median :42.00	Median :36.00
Mean :41.87	Mean :33.58
3rd Qu.:62.25	3rd Qu.:48.25
Max. :83.00	Max. :63.00

# Distribuição de frequência para variáveis quantitativas contínuas: comandos

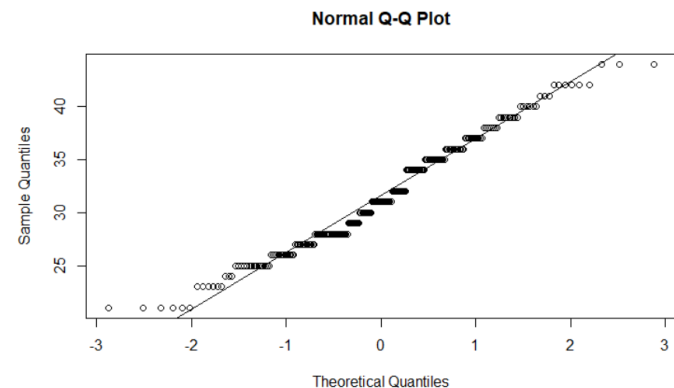
```
> summary(infert$age)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 21.00  28.00  31.00  31.50  35.25  44.00
> sd(infert$age)
[1] 5.251565
> mean(infert$age)
[1] 31.50403
> median(infert$age)
[1] 31
> min(infert$age)
[1] 21
> max(infert$age)
[1] 44
```

## Identificar distribuição normal

```
qqnorm(infert$age)
qqline(infert$age)
shapiro.test(infert$age)
```

Shapiro-Wilk normality test

```
data:  infert$age
W = 0.97606, p-value = 0.0003388
```



# Regressão logística binomial

Relembrando.....

- **y é um evento binário**

$$Prob(y = 1) = \frac{e^{a+xb}}{1 + e^{a+xb}}$$

- 

=

$$\frac{1}{1 + e^{-(a+xb)}}$$

- **Odds**

$$OR(x) = \exp(\beta)$$

## Modelagem → GLM 1 - caso vs. paridade:

O primeiro passo do processo de modelagem é fazer a análise univariada.

Vamos fazer essa análise, primeiramente, para a variável paridade.

```
glm1<-glm(case ~ X1, family=binomial, data = base de dados)
```

Modelos	Fórmula
GLM1	case ~ paridade

```
glm(formula = case ~ parity, family = binomial, data = infert)
```

```
Call: glm(formula = case ~ parity, family = binomial, data = infert)
```

```
Coefficients:
```

```
(Intercept)      parity  
  -0.71868      0.01506
```

```
Degrees of Freedom: 247 Total (i.e. Null); 246 Residual
```

```
Null Deviance: 316.2
```

```
Residual Deviance: 316.2 AIC: 320.2
```





Agora, adicione outras variáveis ao modelo e analise os resultados:

```
glm(formula = case ~ parity+age+education , family = binomial, data = infert)
```

```
Call:  glm(formula = case ~ parity + age + education, family = binomial,
      data = infert)
```

Coefficients:

(Intercept)	parity	age	education6-11yrs	education12+ yrs
-0.847542	0.019070	0.002076	0.046079	0.069988

```
Degrees of Freedom: 247 Total (i.e. Null); 243 Residual
```

```
Null Deviance: 316.2
```

```
Residual Deviance: 316.1 AIC: 326.1
```

# DADOS AMOROSOS.....

- Para demonstrar a Regressão Logística, vamos explorar os dados sobre infidelidade contidos no data frame *Affairs*, que vem com o pacote AER.
  - É preciso instalar este pacote primeiro – `install.packages("AER")`
- Os dados de infidelidade, conhecidos como Assuntos Amorosos, são baseados em uma pesquisa transversal conduzidos por *Psychology Today* (1969).
- Ele contém 9 variáveis coletadas de 601 participantes e inclui dados sobre:
  - quão frequentemente o respondente teve algum caso sexual extraconjugal no último ano, (razão – numérica)
  - sexo, (nominal – fator)
  - idade, (razão – numérica)
  - anos de casado, (razão – numérica)
  - se tem filhos, (nominal – fator)
  - religiosidade (em uma escala de 5 pontos, de 1=anti a 5=muito),
  - educação, (intervalar – numérica -> inteiro)
  - ocupação (classificação de Hollingshead, em uma escala de 7 pontos invertida), (ordinal – categórica -> inteiro)
  - e uma nota (auto avaliação do casamento, de 1=muito infeliz a 5=muito feliz). (intervalar – numérica -> inteiro)

## DADOS AMOROSOS.....

```
> data(Affairs, package = "AER")
```

```
> summary(Affairs)
```

affairs	gender	age	yearsmarried	children
Min. : 0.000	female:315	Min. :17.50	Min. : 0.125	no :171
1st Qu.: 0.000	male :286	1st Qu.:27.00	1st Qu.: 4.000	yes:430
Median : 0.000		Median :32.00	Median : 7.000	
Mean : 1.456		Mean :32.49	Mean : 8.178	
3rd Qu.: 0.000		3rd Qu.:37.00	3rd Qu.:15.000	
Max. :12.000		Max. :57.00	Max. :15.000	

religiousness	education	occupation	rating
Min. :1.000	Min. : 9.00	Min. :1.000	Min. :1.000
1st Qu.:2.000	1st Qu.:14.00	1st Qu.:3.000	1st Qu.:3.000
Median :3.000	Median :16.00	Median :5.000	Median :4.000
Mean :3.116	Mean :16.17	Mean :4.195	Mean :3.932
3rd Qu.:4.000	3rd Qu.:18.00	3rd Qu.:6.000	3rd Qu.:5.000
Max. :5.000	Max. :20.00	Max. :7.000	Max. :5.000

```
> table(Affairs$affairs)
```

0	1	2	3	7	12
451	34	17	19	42	38

# DADOS AMOROSOS.....

```
> str(Affairs, strict.width = "wrap")
'data.frame':  601 obs. of  9 variables:
 $ affairs : num 0 0 0 0 0 0 0 0 0 0 ...
 $ gender  : Factor w/ 2 levels "female","male": 2 1 1 2 2 1 1 2 1 2 ...
 $ age     : num 37 27 32 57 22 32 22 57 32 22 ...
 $ yearsmarried : num 10 4 15 15 0.75 1.5 0.75 15 15 1.5 ...
 $ children : Factor w/ 2 levels "no","yes": 1 1 2 2 1 1 1 2 2 1 ...
 $ religiousness: int 3 4 1 5 2 2 2 2 4 4 ...
 $ education  : num 18 14 12 18 17 17 12 14 16 14 ...
 $ occupation : int 7 6 1 6 6 5 1 4 1 4 ...
 $ rating     : int 4 4 4 5 3 5 3 4 2 5 ...
```

# DADOS AMOROSOS.....

- Podemos ver que 52% dos respondentes são mulheres, que 72% tem filhos e que a idade mediana para amostra foi de 32 anos.
- Com relação à variável resposta, 75% dos respondentes reportaram nenhum engajamento com infidelidade no ano anterior(451/601). O maior número de encontros relatado foi 12 (6%).
- Apesar de ter sido registrado o *número* de traições, nosso interesse é no resultado binário (teve um *affair*/não teve um *afair*).
- Vamos transformar as traições em um fator dicotômico chamado *ynaffair*

```
> Affairs$ynaffair[Affairs$affairs > 0] <- 1
> Affairs$ynaffair[Affairs$affairs == 0] <- 0
> Affairs$ynaffair <- factor(Affairs$ynaffair, levels = c(0, 1),
  labels = c("No", "Yes"))
> table(Affairs$ynaffair)
```

```
  No  Yes
451 150
```

O fator dicotômico pode ser agora utilizado como a variável resposta em um modelo de regressão logística.

# DADOS AMOROSOS.....

```
> fit.full <- glm(yaffair ~ gender + age + yearsmarried + children +
  religiousness + education + occupation + rating, data = Affairs,
  family = binomial())
> summary(fit.full)

Call:
glm(formula = yaffair ~ gender + age + yearsmarried + children +
  religiousness + education + occupation + rating, family = binomial(),
  data = Affairs)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5713  -0.7499  -0.5690  -0.2539   2.5191

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    1.37726    0.88776   1.551 0.120807
gendermale     0.28029    0.23909   1.172 0.241083
age           -0.04426    0.01825  -2.425 0.015301 *
yearsmarried   0.09477    0.03221   2.942 0.003262 **
childrenyes    0.39767    0.29151   1.364 0.172508
religiousness -0.32472    0.08975  -3.618 0.000297 ***
education      0.02105    0.05051   0.417 0.676851
occupation     0.03092    0.07178   0.431 0.666630
rating        -0.46845    0.09091  -5.153 2.56e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 675.38  on 600  degrees of freedom
Residual deviance: 609.51  on 592  degrees of freedom
AIC: 627.51

Number of Fisher Scoring iterations: 4
```

# DADOS AMOROSOS.....

- Olhando os p-value para os coeficientes da regressão (última coluna), vemos que *gênero*, *ter filhos*, *educação*, e *ocupação* podem não fazer uma contribuição significativa para a equação (não podemos rejeitar a hipótese de que os parâmetros são 0).
- O cálculo manual do *Z value* e da  $\Pr(>|z|)$  é feito da seguinte forma:

$$z_{value} = Estimate / Std.Error$$

$$\Pr(z_{value}) = 2 \times (1 - pnorm(z_{value}))$$

```
> # Fazendo para yearsmarried  
> (z_value <- 0.09477302/0.03221445)  
[1] 2.941941  
> (Przvalue <- 2 * (1 - pnorm(z_value)))  
[1] 0.003261618
```

- Vamos ajustar uma segunda equação:

# DADOS AMOROSOS.....

```
> fit.reduced <- glm(yaffair ~ age + yearsmarried + religiousness +  
  rating, data = Affairs, family = binomial())  
> summary(fit.reduced)
```

Call:

```
glm(formula = yaffair ~ age + yearsmarried + religiousness +  
  rating, family = binomial(), data = Affairs)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.6278	-0.7550	-0.5701	-0.2624	2.3998

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )	
(Intercept)	1.93083	0.61032	3.164	0.001558	**
age	-0.03527	0.01736	-2.032	0.042127	*
yearsmarried	0.10062	0.02921	3.445	0.000571	***
religiousness	-0.32902	0.08945	-3.678	0.000235	***
rating	-0.46136	0.08884	-5.193	2.06e-07	***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 675.38 on 600 degrees of freedom  
Residual deviance: 615.36 on 596 degrees of freedom  
AIC: 625.36

Number of Fisher Scoring iterations: 4



# DADOS AMOROSOS.....

- Cada coeficiente de regressão no modelo reduzido é estatisticamente significativo ( $p < .05$ ).
- Como os dois modelos são aninhados (`fit.reduced` é um subconjunto de `fit.full`), podemos utilizar a função `anova()` para compará-los.
- Para modelos lineares generalizados, queremos uma versão *chi-quadrado* do teste:

```
> anova(fit.reduced, fit.full, test = "Chisq")
```

Analysis of Deviance Table

Model 1: `yaffair ~ age + yearsmarried + religiousness + rating`

Model 2: `yaffair ~ gender + age + yearsmarried + children + religiousness + education + occupation + rating`

Resid. Df Resid. Dev Df Deviance Pr(>Chi)

1	596	615.36			
2	592	609.51	4	5.8474	0.2108

# DADOS AMOROSOS.....

- O valor não significativo do qui-quadrado ( $p = 0.21$ ) sugere que o modelo reduzido com 4 preditoras ajusta tão bem como o modelo completo com 9 preditoras,
- Isso reforça nossa hipótese de que gênero, filhos, educação e ocupação não contribuem significativamente para a predição acima e além das outras variáveis na equação.
- Portanto, podemos basear nossa interpretação no modelo mais simples.

## Akaike Information Criteria (AIC)

- O Akaike Information Criterion (AIC) também provê um método para avaliar a qualidade do modelo através de uma comparação de modelos relacionados.
- Ele é baseado nos Desvios, mas impõe uma penalidade para modelos mais complexos.
- Seu objetivo (muito parecido com o **R-quadrado**) é tentar impedir que se inclua preditoras irrelevantes no modelo.
- Contudo, ao contrário do **R-quadrado**, o número em si mesmo não tem significado.
- É preciso ter mais do que um modelo candidato similar (onde todas as variáveis do modelo mais simples ocorrem no modelo mais complexo), **então escolhemos o modelo que tem o menor AIC.**

# DADOS AMOROSOS.....

Vamos olhar os coeficientes da regressão:

```
> coef(fit.reduced)
(Intercept)          age  yearsmarried religiousness      rating
  1.93083017   -0.03527112    0.10062274   -0.32902386   -0.46136144
```

- Em uma regressão logística, a resposta sendo modelada é o  $\log(odds)$  de que  $Y = 1$ .
- Os coeficientes da regressão nos dão a mudança em  $\log(odds)$  na resposta para uma unidade de mudança na variável preditora, mantendo-se todas as outras preditoras constantes.
- Como  $\log(odds)$  são difíceis de se interpretar, podemos exponenciá-lo e colocar os resultados em uma escala de *chances* (*odds*):


```
> exp(coef(fit.reduced))
(Intercept)          age  yearsmarried religiousness      rating
  6.8952321    0.9653437    1.1058594    0.7196258    0.6304248
```

## INTERPRETAÇÃO - DADOS AMOROSOS.....

- Podemos ver que as chances de um encontro extraconjugal aumentam por um fator de 1.106 para um aumento de um ano em *anos de casado* (mantendo-se idade, religiosidade e avaliação conjugal constante).
- Inversamente, as chances de um caso extraconjugal são multiplicadas por um fator de 0.965 para cada ano de aumento na idade. As chances de um encontro extraconjugal aumentam com os anos de casamento e diminuem com a idade, religiosidade e avaliação conjugal.
- Como as variáveis preditoras não podem ser iguais a 0, o deslocamento (intercept) não é significativo neste caso.
- Se for desejável, pode-se obter os intervalos de confiança dos coeficientes com a função `confint()`.

# INTERPRETAÇÃO - DADOS AMOROSOS.....

- Finalmente, pode ser que não estejamos interessados em uma mudança de uma unidade em uma variável preditora.
- Por exemplo, para regressão logística binária, a mudança nas chances de um valor maior na variável resposta para uma mudança de  $n$  unidades em uma variável preditora é  $\exp(\beta_j)^n$ .
- Se um ano de aumento em *anos de casado* multiplica as chances de um caso por 1.106, um aumento de 10 anos aumentaria as chances por um fator de  $1.106^{10} = 2.7$ , mantendo-se todas as outras variáveis preditoras constantes.



## Avaliando-se o impacto das preditoras na probabilidade de um resultado

- Geralmente, é mais fácil pensar em termos de probabilidades do que de chances.
- Podemos utilizar a função `predict()` para observar o impacto da variação dos níveis das variáveis preditoras na probabilidade de um resultado.
- O primeiro passo é criar um conjunto de dados artificial contendo os valores das variáveis preditoras que estamos interessado.
- Então podemos utilizar este conjunto de dados artificial com a função `predict()` para prever as probabilidades do evento resultado ocorrer para estes valores.
- Vamos aplicar esta estratégia para avaliar o impacto da avaliação conjugal sobre a probabilidade de se ter um caso extraconjugal.
- Primeiro criamos um conjunto de dados artificial onde idade, anos casado e religiosidade são colocados como seus valores médios, e a avaliação conjugal varia de 1 a 5.

## Avaliando-se o impacto das preditoras

```
> testdata <- data.frame(rating=c(1,2,3,4,5), age=mean(Affairs$age), #
                        yearsmarried=mean(Affairs$yearsmarried), #
                        religiousness=mean(Affairs$religiousness))

> testdata
  rating    age yearsmarried religiousness
1      1 32.48752      8.177696      3.116473
2      2 32.48752      8.177696      3.116473
3      3 32.48752      8.177696      3.116473
4      4 32.48752      8.177696      3.116473
5      5 32.48752      8.177696      3.116473
```

- Agora vamos usar este conjunto de dados de teste e a equação de previsão para obter as probabilidades:

```
> testdata$prob <- predict(fit.reduced, newdata = testdata, type = "response")
> testdata
  rating    age yearsmarried religiousness    prob
1      1 32.48752      8.177696      3.116473 0.5302296
2      2 32.48752      8.177696      3.116473 0.4157377
3      3 32.48752      8.177696      3.116473 0.3096712
4      4 32.48752      8.177696      3.116473 0.2204547
5      5 32.48752      8.177696      3.116473 0.1513079
```



## Avaliando-se o impacto das preditoras

- Destes resultados vemos que a probabilidade de um caso extraconjugal diminuiu de 0.53 quando o casamento é avaliado em 1=muito infeliz para 0.15 quando o o casamento é avaliado em 5=muito feliz (mantendo-se idade, anos casado, e religiosidade contantes).
- Agora vamos ver o impacto da idade.

```
> testdata <- data.frame(rating=mean(Affairs$rating), age=seq(17,57,10), #
                        yearsmarried=mean(Affairs$yearsmarried), #
                        religiousness=mean(Affairs$religiousness))
> testdata$prob <- predict(fit.reduced, newdata=testdata, type="response")
> testdata
```

	rating	age	yearsmarried	religiousness	prob
1	3.93178	17	8.177696	3.116473	0.3350834
2	3.93178	27	8.177696	3.116473	0.2615373
3	3.93178	37	8.177696	3.116473	0.1992953
4	3.93178	47	8.177696	3.116473	0.1488796
5	3.93178	57	8.177696	3.116473	0.1094738





## Avaliando-se o impacto das preditoras

- Podemos ver que conforme a idade aumenta de 17 a 57, a probabilidade de um encontro extraconjugal diminuir de 0.34 a 0.11, mantendo-se as outras variáveis constantes.
- Utilizando-se esta abordagem, podemos explorar o impacto de cada uma das variáveis preditoras no resultado.



**Extras**

## Super Dispersão (*Overdispersion*)

- A variância esperada para os dados obtidos de uma distribuição binomial é  $\sigma^2 = n\pi(1 - \pi)$ , onde  $n$  é o número de observações e  $\pi$  é a probabilidade de se pertencer ao grupo  $Y = 1$ .
- Super Dispersão (*Overdispersion*) ocorre quando a variância observada da variável resposta é maior do que seria esperado de uma distribuição binomial.
- Super Dispersão pode levar a testes distorcidos de erros padrões e testes imprecisos de significância.
- Quando se tem super dispersão, o ajuste com uma função logística ainda é possível utilizando-se a função `glm()`, mas neste caso, é preciso utilizar a distribuição *quasibinomial* ao invés da distribuição binomial.

## Super Dispersão (*Overdispersion*)

- Uma maneira de se detectar a super dispersão é comparar o desvio residual com os graus de liberdade dos resíduos no nosso modelo binomial. Se a razão

$$\phi = \frac{\text{Desvio Residual}}{\text{GL do Residuo}}$$

é consideravelmente maior do que 1, temos evidência de super dispersão.

- Aplicando isso ao exemplo dos dados Affairs temos:

```
> deviance(fit.reduced)/df.residual(fit.reduced)
[1] 1.03248
```

que é muito próximo de 1, sugerindo que não temos super dispersão.

## Super Dispersão (*Overdispersion*)

- Outro teste que podemos fazer para verificar se temos ou não super dispersão é ajustar o modelo duas vezes:
  - Na primeira vez utilizamos `family="binomial"`
  - Na segunda vez utilizamos `family="quasibinomial"`
- Se o objeto `glm()` retornado no primeiro caso é `fit` e o objeto retornado no segundo caso é `fit.od`, então fazemos:

```
> pchisq(summary(fit.od)$dispersion * fit$df.residual, fit$df.residual,  
lower = F)
```

- O resultado da função acima é o *p-value* para testarmos a hipótese nula  $H_0 : \phi = 1$  versus a hipótese alternativa  $H_1 : \phi \neq 1$ .
- Se  $p$  é pequeno (ou seja, menor que 0.05), rejeitamos a hipótese nula.

## Super Dispersão (*Overdispersion*)

- Aplicando isto ao conjunto de dados Affairs, temos:

```
> fit <- glm(yaffair ~ age + yearsmarried + religiousness + rating, #  
            family = binomial(), data=Affairs)  
> fit.od <- glm(yaffair ~ age + yearsmarried + religiousness + rating, #  
               family = quasibinomial(), data=Affairs)  
> pchisq(summary(fit.od)$dispersion * fit$df.residual, fit$df.residual, lower=F)  
[1] 0.340122
```

- O resultado do *p-value* (0.34) é claramente não significativo ( $p > 0.05$ ), fortalecendo nossa crença de que super dispersão não é um problema (não podemos rejeitar a hipótese nula de que  $H_0 : \phi = 1$ ).

## Voltando ao exemplo do Titanic....

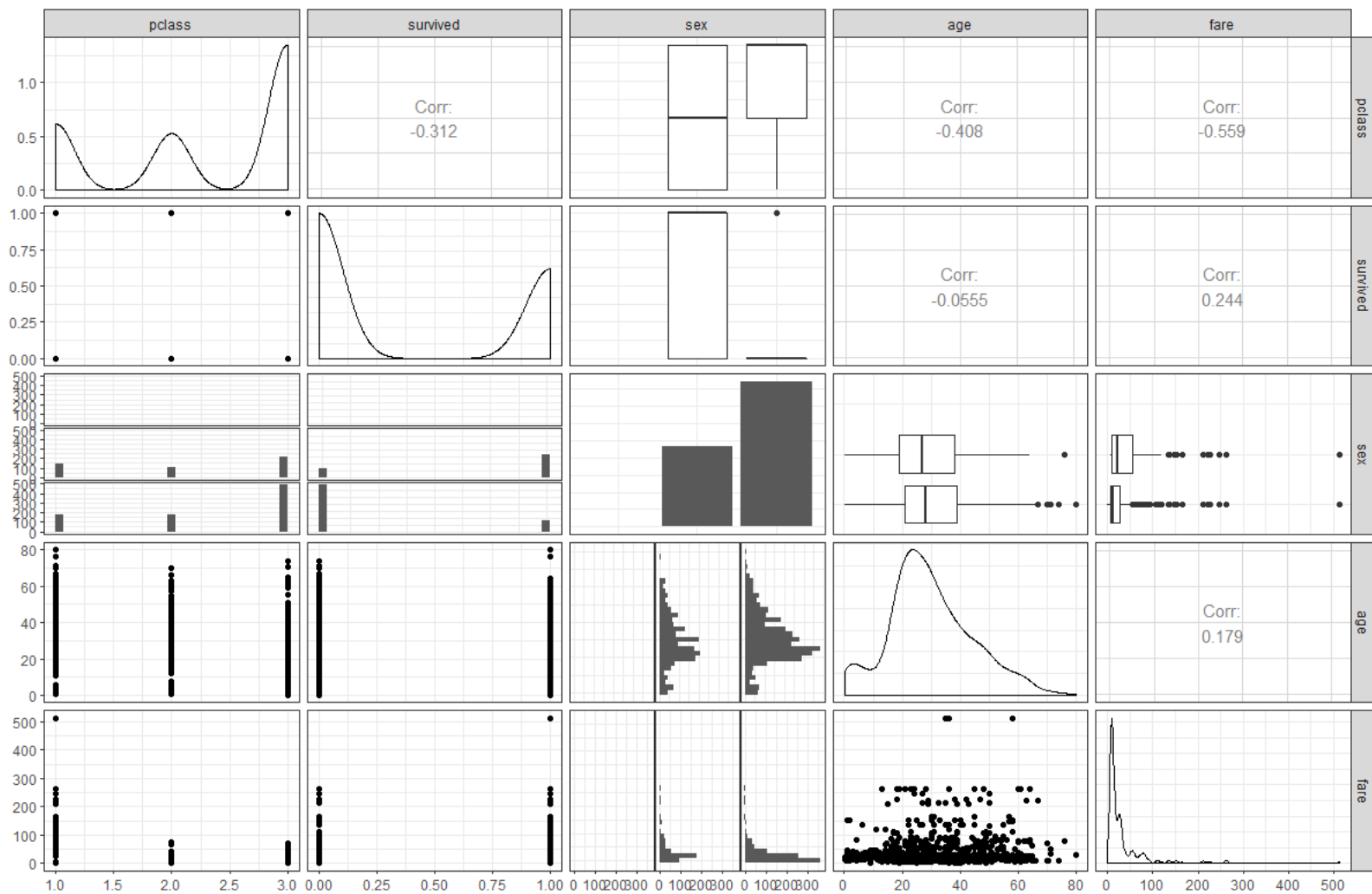
Baixe o conjunto de dados titanic.txt. Cada observação deste banco é relativa a um passageiro do Titanic.

- As covariáveis indicam características destes passageiros;
- a variável resposta indica se o passageiro sobreviveu ou não ao naufrágio.

```
titanic<- read.csv("C:/Users/Unicsul/Multivariada II/titanic3.csv", header = TRUE, sep = ";", dec=",")
```

```
▼ titanic      1310 obs. of 14 variables
 pclass : int 1 1 1 1 1 1 1 1 1 1 ...
 survived : int 1 1 0 0 0 1 1 0 1 0 ...
 name : Factor w/ 1308 levels "", "Abbing, Mr. Anthony",...: 23 25 26 27 28 32 47 4
 sex : Factor w/ 3 levels "", "female", "male": 2 3 2 3 2 3 2 3 2 3 ...
 age : num 29 0.917 2 30 25 ...
 sibsp : int 0 1 1 1 1 0 1 0 2 0 ...
 parch : int 0 2 2 2 2 0 0 0 0 0 ...
 ticket : Factor w/ 930 levels "", "110152", "110413",...: 189 51 51 51 51 126 94 17
 fare : num 211 152 152 152 152 ...
 cabin : Factor w/ 187 levels "", "A10", "A11",...: 45 81 81 81 81 151 147 17 63 1 .
 embarked : Factor w/ 4 levels "", "C", "Q", "S": 4 4 4 4 4 4 4 4 4 2 ...
 boat : Factor w/ 28 levels "", "1", "10", "11",...: 13 4 1 1 1 14 3 1 28 1 ...
 body : int NA NA NA 135 NA NA NA NA NA 22 ...
 home.dest: Factor w/ 370 levels "", "?Havana, Cuba",...: 310 232 232 232 232 238 1
```

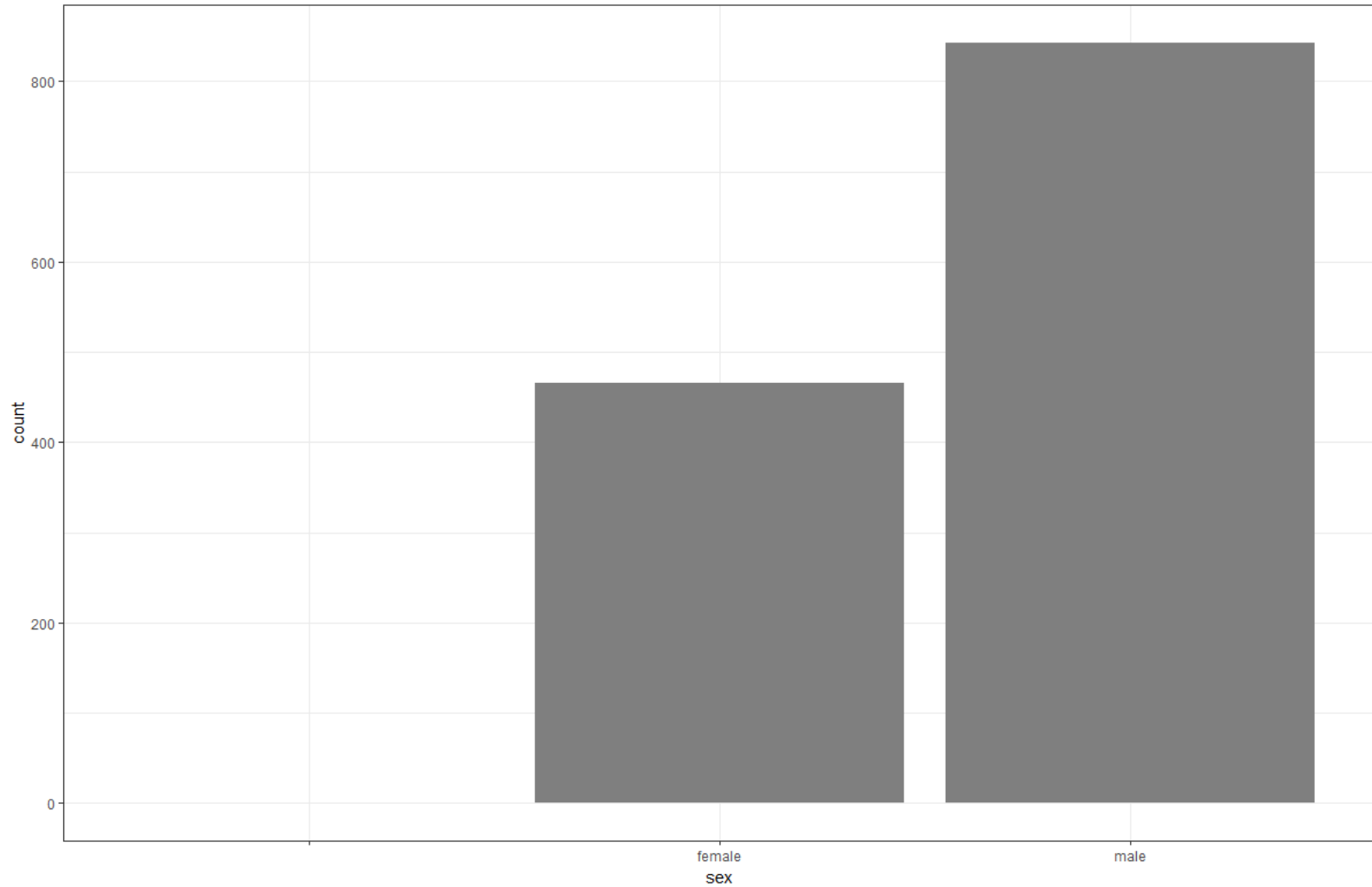
```
ggpairs(titanic)
```





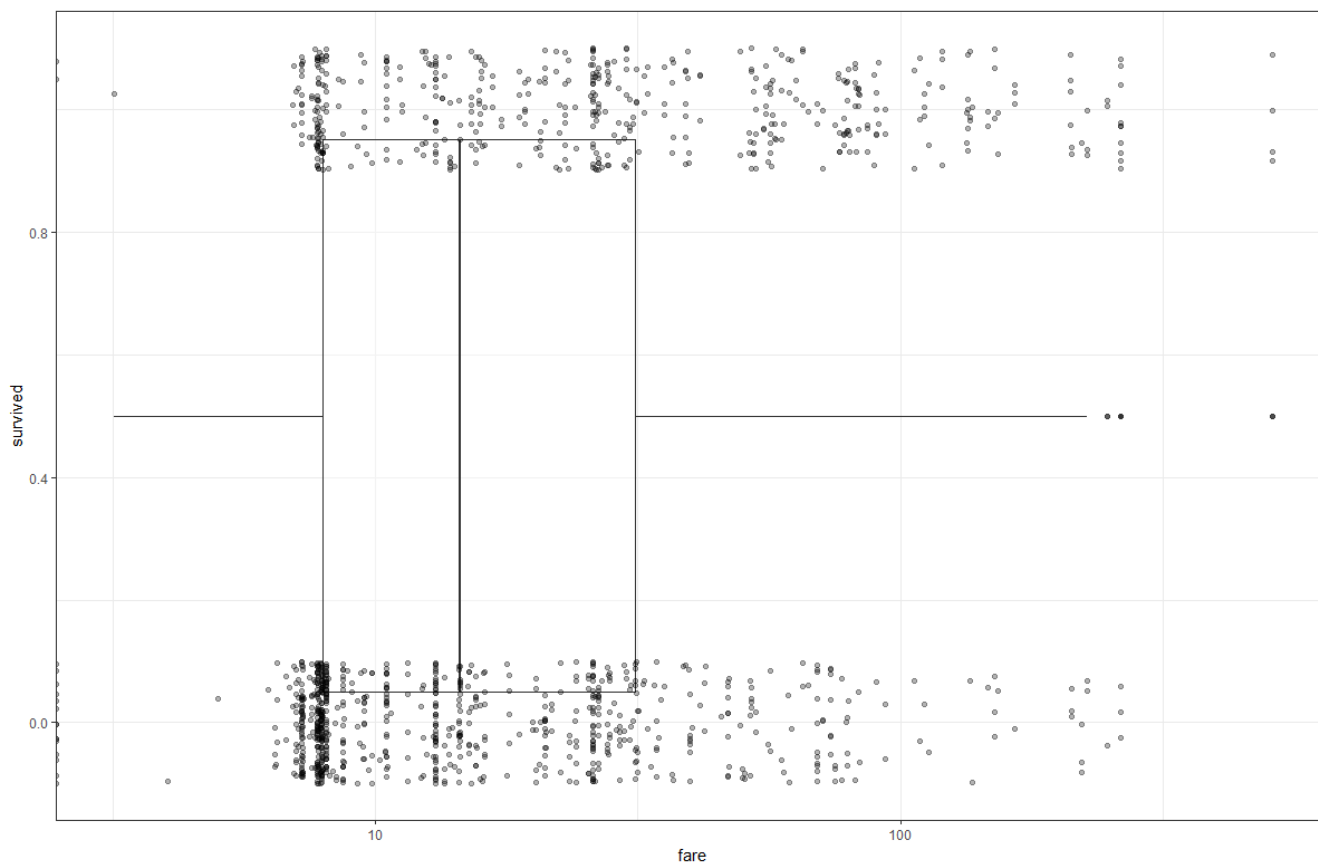


```
titanic %>%  
  ggplot(aes(x = sex, fill = survived)) +  geom_bar(position = "dodge")
```



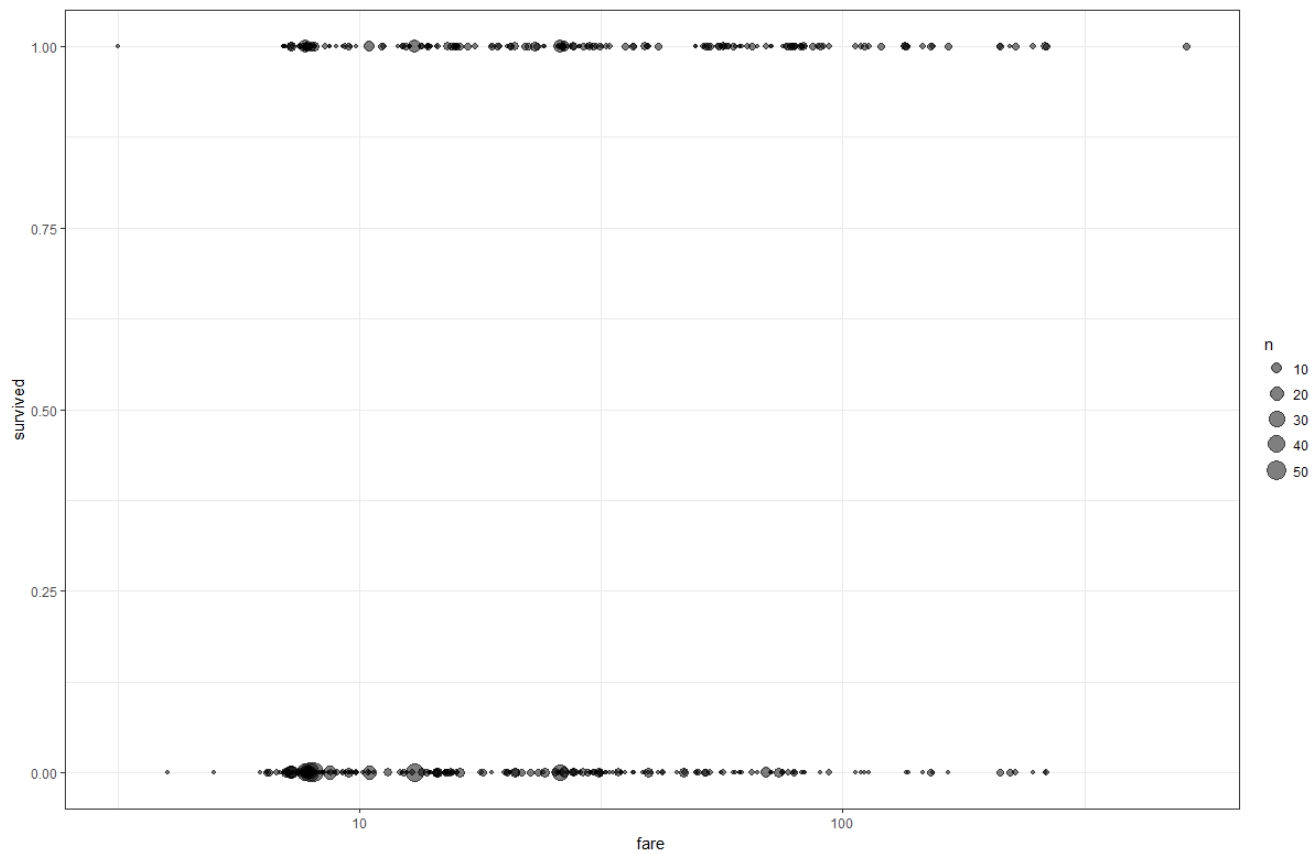
Parece haver uma relação entre fare e survived.....

```
ggplot(titanic, aes(x = survived, y = fare)) +  
  #geom_violin(aes(fill = survived), alpha = .4) +  
  geom_boxplot(aes(fill = survived), alpha = .4) +  
  #geom_count() +  
  geom_jitter(width = .1, alpha = .3) + coord_flip() + scale_y_log10()
```



Seria possível passar uma regressão linear?

```
titanic %>%  
  filter(fare > 0) %>%  
  ggplot(aes(x = fare, y = survived)) + scale_x_log10() + geom_count(alpha = .5)
```



# Fit univariado no exemplo com o Titanic

A interpretação é semelhante à regressão linear. Exceto que os valores dos coeficientes sem o exp fazem pouco sentido. Aqui é melhor usar a noção de odds ratio. Para isso basta exponenciar os coeficientes encontrados.

```
titanic_t = titanic %>%
  filter(fare > 0) %>%
  mutate(logFare = log(fare), # cria ou modifica colunas
         survived = as.factor(survived))
# glm que usaremos abaixo lida melhor com factor que character
```

```
bm <- glm(survived ~ logFare, data = titanic_t, family = "binomial")
tidy(bm, conf.int = TRUE)
```

```
> tidy(bm, conf.int = TRUE)
# A tibble: 2 x 7
  term          estimate std.error statistic  p.value conf.low conf.high
<chr>         <dbl>     <dbl>     <dbl>   <dbl>   <dbl>   <dbl>
1 (Intercept)  -2.49         0.205     -12.2 4.49e-34 -2.90   -2.10
2 logFare       0.679         0.0652     10.4 2.21e-25  0.552   0.808
```

# EXPONENCIANDO:

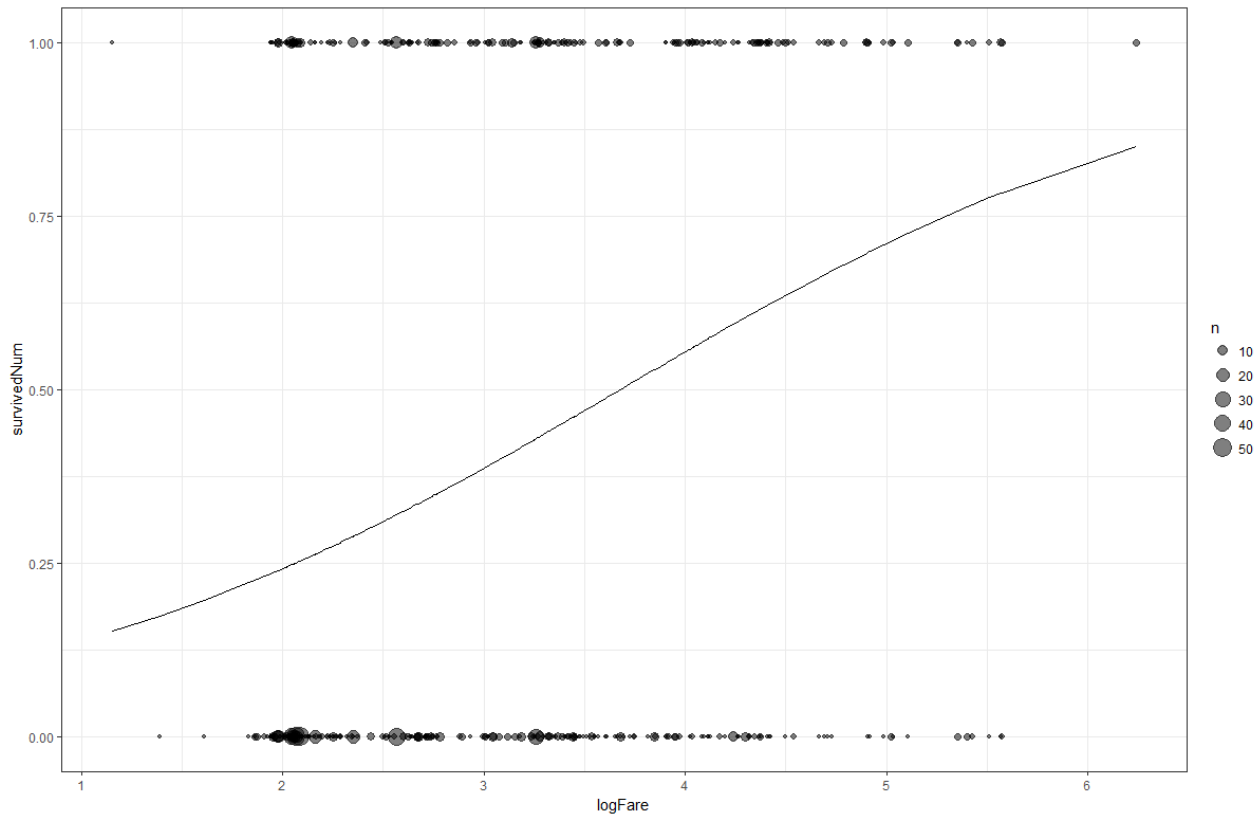
```
tidy(bm, conf.int = TRUE, exponentiate = TRUE)
```

```
> tidy(bm, conf.int = TRUE, exponentiate = TRUE)
# A tibble: 2 x 7
  term          estimate std.error statistic  p.value conf.low conf.high
<chr>         <dbl>     <dbl>     <dbl>   <dbl>   <dbl>   <dbl>
1 (Intercept)  0.0825         0.205     -12.2 4.49e-34  0.0549  0.123
2 logFare       1.97         0.0652     10.4 2.21e-25  1.74    2.24
```

## Como aqui  $y = \exp(b_0) \cdot \exp(b_1 \cdot x_1)$ , aumentar em uma unidade  $x$ , faz com que  $y$  seja multiplicado por  $\exp(b_1)$ , que é o coeficiente acima

## Visualizando o modelo...

```
bm %>%  
  augment(type.predict = "response") %>%  
  mutate(survivedNum = ifelse(survived == "1", 1, 0)) %>%  
  ggplot(aes(x = logFare)) +  
  geom_count(aes(y = survivedNum), alpha = 0.5) +  
  geom_line(aes(y = .fitted))
```



## Preditor categórico

```
bm <- glm(survived ~ pclass, data = titanic_t, family = "binomial")
tidy(bm, conf.int = TRUE)
glance(bm) # Os métodos glance sempre retornam um quadro de dados de uma linha (exceto em
NULL, que retorna um quadro de dados vazio)
```

```
# A tibble: 2 x 7
  term          estimate std.error statistic  p.value conf.low conf.high
<chr>         <dbl>     <dbl>     <dbl>   <dbl>   <dbl>   <dbl>
1 (Intercept)    1.34      0.170      7.87 3.58e-15    1.01    1.68
2 pclass        -0.802    0.0717    -11.2 5.37e-29   -0.943   -0.662
> glance(bm)
# A tibble: 1 x 7
  null.deviance df.null logLik  AIC  BIC deviance df.residual
      <dbl>     <int>  <dbl> <dbl> <dbl>   <dbl>     <int>
1      1722.     1290  -794. 1593. 1603.   1589.     1289
```

```
#summary(bm)
```

```
Call:
glm(formula = survived ~ pclass, family = "binomial", data = titanic_t)
```

```
Deviance Residuals:
```

```
    Min       1Q   Median       3Q      Max
-1.4132  -0.7700  -0.7700   0.9586   1.6496
```

```
Coefficients:
```

```
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.34080    0.17040   7.869 3.58e-15 ***
pclass      -0.80164    0.07173  -11.176 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

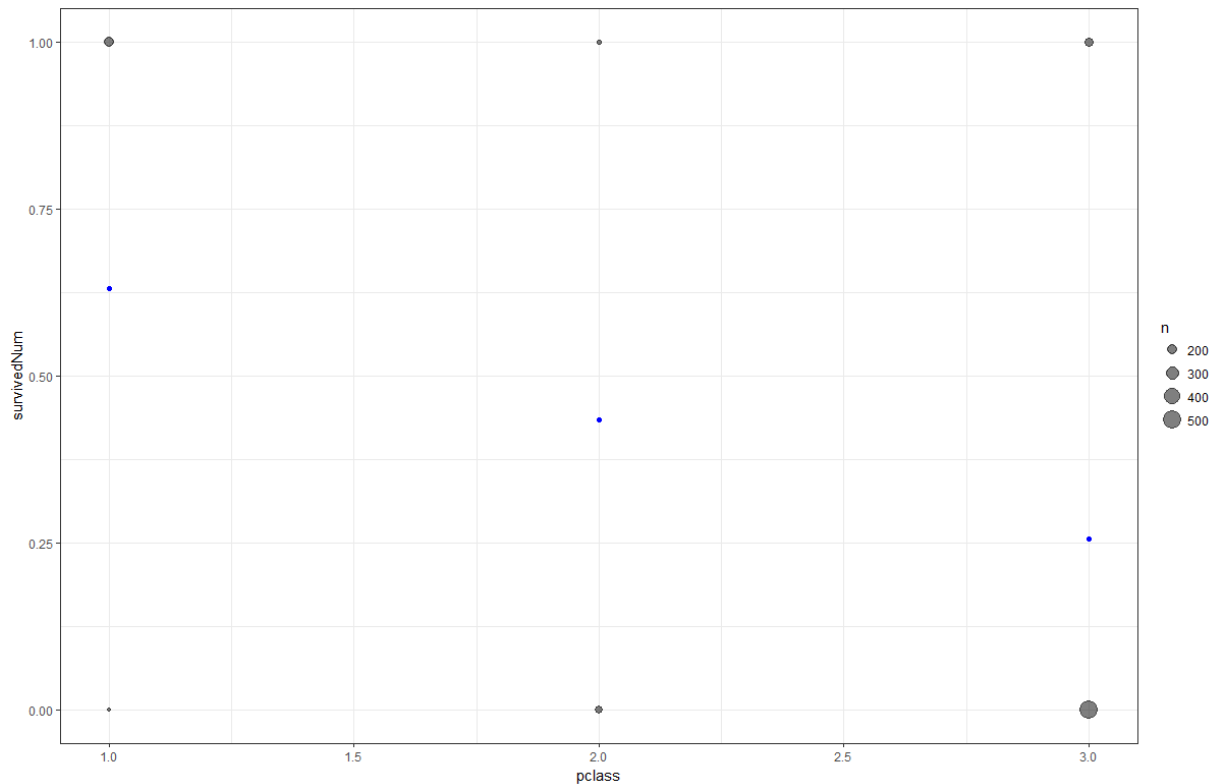
```
(Dispersion parameter for binomial family taken to be 1)
```

```
Null deviance: 1721.7 on 1290 degrees of freedom
Residual deviance: 1588.7 on 1289 degrees of freedom
AIC: 1592.7
```

```
Number of Fisher Scoring iterations: 4
```

## Preditor categórico

```
bm %>%  
  augment(type.predict = "response") %>%  
  mutate(survivedNum = ifelse(survived == "1", 1, 0)) %>%  
  ggplot(aes(x = pclass)) +  
  geom_count(aes(y = survivedNum), alpha = 0.5) +  
  geom_point(aes(y = .fitted), color = "blue")
```



## Multivariada:

```
bm <- glm(survived ~ pclass + sex + age + sex*age, data = titanic_t, family = "binomial")
```

```
tidy(bm, conf.int = TRUE)
```

```
tidy(bm, conf.int = TRUE, exponentiate = TRUE)
```

```
glance(bm)
```

```
> tidy(bm, conf.int = TRUE)
```

```
# A tibble: 5 x 7
```

	term <chr>	estimate <dbl>	std.error <dbl>	statistic <dbl>	p.value <dbl>	conf.low <dbl>	conf.high <dbl>
1	(Intercept)	3.89	0.438	8.87	7.14e-19	3.04	4.76
2	pclass	-1.18	0.115	-10.2	2.32e-24	-1.41	-0.954
3	sexmale	-1.07	0.359	-2.98	2.84e- 3	-1.78	-0.370
4	age	-0.00438	0.00948	-0.462	6.44e- 1	-0.0228	0.0144
5	sexmale:age	-0.0513	0.0119	-4.30	1.74e- 5	-0.0751	-0.0282

```
> tidy(bm, conf.int = TRUE, exponentiate = TRUE)
```

```
# A tibble: 5 x 7
```

	term <chr>	estimate <dbl>	std.error <dbl>	statistic <dbl>	p.value <dbl>	conf.low <dbl>	conf.high <dbl>
1	(Intercept)	48.7	0.438	8.87	7.14e-19	21.0	117.
2	pclass	0.309	0.115	-10.2	2.32e-24	0.245	0.385
3	sexmale	0.342	0.359	-2.98	2.84e- 3	0.169	0.691
4	age	0.996	0.00948	-0.462	6.44e- 1	0.977	1.01
5	sexmale:age	0.950	0.0119	-4.30	1.74e- 5	0.928	0.972

```
> glance(bm)
```

```
# A tibble: 1 x 7
```

	null.deviance <dbl>	df.null <int>	logLik <dbl>	AIC <dbl>	BIC <dbl>	deviance <dbl>	df.residual <int>
1	1404.	1036	-476.	963.	988.	953.	1032



## Avaliando o modelo pela precisão:

```
bm %>% augment(type.predict = "response")

predictions <- predict(bm, type = "response") > .5
titanic_t = titanic_t %>% mutate(true_survivals = survived == 1)
```

```
titanic_t
```

```
> titanic_t
```

	pclass	survived	sex	age	fare	logFare	true_survivals
1	1	1	female	29.0000	211.3375	5.353456	TRUE
2	1	1	male	0.9167	151.5500	5.020916	TRUE
3	1	0	female	2.0000	151.5500	5.020916	FALSE
4	1	0	male	30.0000	151.5500	5.020916	FALSE
5	1	0	female	25.0000	151.5500	5.020916	FALSE
6	1	1	male	48.0000	26.5500	3.279030	TRUE
7	1	1	female	63.0000	77.9583	4.356174	TRUE
8	1	1	female	53.0000	51.4792	3.941178	TRUE
9	1	0	male	71.0000	49.5042	3.902058	FALSE
10	1	0	male	47.0000	227.5250	5.427260	FALSE
11	1	1	female	18.0000	227.5250	5.427260	TRUE
12	1	1	female	24.0000	69.3000	4.238445	TRUE
13	1	1	female	26.0000	78.8500	4.367547	TRUE
14	1	1	male	80.0000	30.0000	3.401197	TRUE

```
erro <- sum((predictions != titanic_t$true_survivals)) / NROW(predictions)
erro
```

```
> erro
[1] 0.5901639
```



# Credit Score usando o R

Credit Scoring é definido como sendo um modelo estatístico/econométrico o qual atribui uma medida de **risco** aos clientes de uma instituição financeira ou aos futuros clientes.

Usualmente, a avaliação dos clientes é realizada por meio de um Credit Scorecard o qual é um modelo estatístico para avaliação do risco estruturado de maneira a facilitar a tomada de decisão quanto a liberação do crédito ou não.

O uso de credit scorecard é muito popular principalmente para as organizações que lidam empréstimos como bancos.

Dentre as vantagens da utilização de um credit scorecard podemos listar:

Credit Scorecard é implementado facilmente e pode ser monitorado ao longo do tempo. Pessoas sem o conhecimento técnico em estatística ou econometria podem utilizar facilmente o credit scorecard para tomar decisões.

As principais questões em Credit Score são:

Quem receberá o crédito ?

Quanto deverá ser esse crédito ?

Quais as estratégias para a distribuição e cobrança do crédito ?



# Credit Score usando o R

Para demonstrar como o [Credit Score](#) pode ser formulado usando o R, iremos trabalhar com os dados [German.csv](#) o qual representa um [conjunto de dados de crédito para uma instituição financeira Alemã](#).

Os dados são compostos por 300 empréstimos "*ruins*" (por exemplo, ausência de pagamento ou atraso) e 700 empréstimos "*bons*" (por exemplo, pagamentos sem atraso).

O objetivo é fornecer insumos para a tomada de decisão quanto aos futuros empréstimos com base nos padrões anteriormente observados.

Vamos importar os dados para o R:

```
#Limpa o Workspace  
rm(list=ls())
```

```
#Importa os dados German.csv  
dados.df<-read.csv("C:/Users/Multivariada II/GermanCredit.csv", header = TRUE, sep = ";",  
dec=",")
```

```
#Apresenta as variáveis do DataFrame  
names(dados.df)
```

```
#Apresenta a estrutura do DataFrame  
str(dados.df)
```

# Credit Score usando o R

No R as variáveis categóricas ou binárias são usualmente tratadas como **fatores** enquanto as variáveis contínuas ou discretas são tratadas como valores **numéricos**. Nesse caso, algumas conversões são necessárias:

```
#Transforma em fatores as variáveis categóricas e "dummies"
dados.df[, "CHK_ACCT"]      <-as.factor(dados.df[, "CHK_ACCT"])
dados.df[, "HISTORY"]       <-as.factor(dados.df[, "HISTORY"])
dados.df[, "NEW_CAR"]       <-as.factor(dados.df[, "NEW_CAR"])
dados.df[, "USED_CAR"]      <-as.factor(dados.df[, "USED_CAR"])
dados.df[, "FURNITURE"]     <-as.factor(dados.df[, "FURNITURE"])
dados.df[, "RADIO_TV"]      <-as.factor(dados.df[, "RADIO_TV"])
dados.df[, "EDUCATION"]     <-as.factor(dados.df[, "EDUCATION"])
dados.df[, "RETRAINING"]    <-as.factor(dados.df[, "RETRAINING"])
dados.df[, "SAV_ACCT"]      <-as.factor(dados.df[, "SAV_ACCT"])
dados.df[, "EMPLOYMENT"]    <-as.factor(dados.df[, "EMPLOYMENT"])
dados.df[, "MALE_DIV"]      <-as.factor(dados.df[, "MALE_DIV"])
dados.df[, "MALE_SINGLE"]   <-as.factor(dados.df[, "MALE_SINGLE"])
dados.df[, "CO.APPLICANT"]  <-as.factor(dados.df[, "CO.APPLICANT"])
dados.df[, "GUARANTOR"]     <-as.factor(dados.df[, "GUARANTOR"])
dados.df[, "REAL_ESTATE"]   <-as.factor(dados.df[, "REAL_ESTATE"])
dados.df[, "OTHER_INSTALL"] <-as.factor(dados.df[, "OTHER_INSTALL"])
dados.df[, "RENT"]          <-as.factor(dados.df[, "RENT"])
dados.df[, "OWN_RES"]       <-as.factor(dados.df[, "OWN_RES"])
dados.df[, "NUM_CREDITS"]    <-as.factor(dados.df[, "NUM_CREDITS"])
dados.df[, "JOB"]           <-as.factor(dados.df[, "JOB"])
dados.df[, "TELEPHONE"]     <-as.factor(dados.df[, "TELEPHONE"])
dados.df[, "FOREIGN"]       <-as.factor(dados.df[, "FOREIGN"])
```

# Credit Score usando o R

```
#Variável dependente
dados.df[, "RESPONSE"]      <-as.factor(dados.df[, "RESPONSE"])

#Transforma em numeric
dados.df[, "AMOUNT"]        <-as.numeric(dados.df[, "AMOUNT"])
dados.df[, "INSTALL_RATE"]  <-as.numeric(dados.df[, "INSTALL_RATE"])
dados.df[, "AGE"]           <-as.numeric(dados.df[, "AGE"])
dados.df[, "DURATION"]      <-as.numeric(dados.df[, "DURATION"])
```

O próximo passo é separar os dados em dois grupos:

Dados para estimação. (Treinamento)

Dados para teste. (Validação)

Uma sugestão é separar a base de dados (aleatoriamente) da seguinte forma: 60% das observações deverão compor a base de treinamento e 40% a base de validação:

```
#Índices obtidos após a aleatorização
ordena <- sort(sample(nrow(dados.df), nrow(dados.df) *.6))

#Dados para o treinamento
treinamento<-dados.df[ordena,]

#Dados para a validação
validacao<-dados.df[-ordena,]
```

# Credit Score usando o R

A ideia é construir o(s) modelo(s) de [Credit Scoring](#) com o *DataFrame* "treinamento" e em seguida avaliar o ajuste com o *DataFrame* "validacao". Uma das formas mais simples para modelar os dados de crédito é por meio da [regressão logística](#):

```
#Regressão Logística
modelo.completo <- glm(RESPONSE ~ . ,family=binomial,data=treinamento)
```

Como há muitas possíveis variáveis para o modelo, podemos proceder com a abordagem de [Stepwise](#) para selecionar o modelo com a "melhor" combinação de [variáveis explicativas](#):

```
#Abordagem Stepwise para seleção de variáveis
stepwise <- step(modelo.completo,direction="both")
```

(Vai aparecer um zilhão de iterações.....)

Após algumas iterações, observa-se que o conjunto de variáveis com o menor valor para o [Critério de Informação de Akaike](#) é:

```
#Modelo com as variáveis indicadas pelo Stepwise
stepwise <- glm(RESPONSE ~
JOB+NUM_CREDITS+EMPLOYMENT+RETRAINING+NEW_CAR+TELEPHONE+MALE_DIV+
FURNITURE+RENT+REAL_ESTATE+EDUCATION+FOREIGN,
family=binomial,data=treinamento)

#Resume os resultados do modelo
summary(stepwise)
```

# Credit Score usando o R

```
Call:
glm(formula = RESPONSE ~ JOB + NUM_CREDITS + EMPLOYMENT + RETRAINING +
    NEW_CAR + TELEPHONE + MALE_DIV + FURNITURE + RENT + REAL_ESTATE +
    EDUCATION + FOREIGN, family = binomial, data = treinamento)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.4062	-1.2119	0.6732	0.8780	1.4537

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	0.62588	0.63613	0.984	0.32516
JOB1	-0.21522	0.70754	-0.304	0.76099
JOB2	-0.03962	0.68474	-0.058	0.95386
JOB3	-0.30610	0.68968	-0.444	0.65717
NUM_CREDITS2	0.15607	0.20309	0.768	0.44221
NUM_CREDITS3	0.54739	0.70104	0.781	0.43491
NUM_CREDITS4	-0.31784	1.47323	-0.216	0.82919
EMPLOYMENT1	-0.02861	0.44794	-0.064	0.94908
EMPLOYMENT2	0.33528	0.43930	0.763	0.44533
EMPLOYMENT3	0.72834	0.46483	1.567	0.11714
EMPLOYMENT4	0.71843	0.44491	1.615	0.10636
RETRAINING1	-0.45053	0.32965	-1.367	0.17172
NEW_CAR1	-0.67209	0.23672	-2.839	0.00452 **
TELEPHONE1	0.17491	0.21363	0.819	0.41292
MALE_DIV1	-0.50740	0.38499	-1.318	0.18752
FURNITURE1	-0.22893	0.26589	-0.861	0.38924
RENT1	-0.45564	0.24305	-1.875	0.06084 .
REAL_ESTATE1	0.61365	0.22631	2.712	0.00670 **
EDUCATION1	-0.90632	0.38435	-2.358	0.01837 *
FOREIGN	2.14542	1.05143	2.040	0.04130 *

---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 739.69 on 599 degrees of freedom  
Residual deviance: 691.94 on 580 degrees of freedom  
AIC: 731.94

Percebemos que nem todas as variáveis são significantes....

# Credit Score usando o R

Uma medida interessante para interpretar o modelo é a medida de Razão de chances (Odds Ratio):

```
#Calcula a razão de chances  
exp(cbind(OR = coef(stepwise), confint(stepwise)))
```

```
> #Calcula a razão de chances  
> exp(cbind(OR = coef(stepwise), confint(stepwise)))  
Waiting for profiling to be done...
```

	OR	2.5 %	97.5 %
(Intercept)	1.8698995	0.54351433	6.9212403
JOB1	0.8063664	0.19099679	3.1923440
JOB2	0.9611532	0.23723514	3.6363349
JOB3	0.7363128	0.18056450	2.8126394
NUM_CREDITS2	1.1689085	0.78738344	1.7476837
NUM_CREDITS3	1.7287331	0.48514479	8.2626560
NUM_CREDITS4	0.7277166	0.02654571	19.8630351
EMPLOYMENT1	0.9717999	0.40003985	2.3373763
EMPLOYMENT2	1.3983365	0.58432665	3.3020038
EMPLOYMENT3	2.0716416	0.82769774	5.1664979
EMPLOYMENT4	2.0512073	0.84936697	4.9052015
RETRAINING1	0.6372903	0.33680719	1.2331482
NEW_CAR1	0.5106420	0.32070823	0.8123040
TELEPHONE1	1.1911445	0.78558150	1.8173485
MALE_DIV1	0.6020567	0.28395162	1.2998895
FURNITURE1	0.7953847	0.47406879	1.3473600
RENT1	0.6340427	0.39455337	1.0253922
REAL_ESTATE1	1.8471663	1.19374178	2.9035031
EDUCATION1	0.4040067	0.19012465	0.8657673
FOREIGN	8.5456560	1.64235195	157.4130053

Interpretação:

Veja por exemplo a variável *TELEPHONE\_1*, nesse caso, para cada telefone a mais que um proponente possui, isso aumenta a sua chance de ser considerado inadimplente em aproximadamente 19%.

Finalmente, vamos testar a qualidade do modelo aplicando o modelo estimado na base de validação para termos uma ideia do grau de acerto desse modelo:



# Credit Score usando o R

```
#Faz a previsão para a base de validação (probabilidade)
predito<-predict(stepwise,validacao,type="response")
```

```
#Escolhe quem vai ser "1" e quem vai ser "0"
predito<-ifelse(predito>=0.8,1,0)
```

```
#Compara os resultados
MC=table(predito,validacao$RESPONSE)
Show(MC)
```

```
> table(predito,validacao$RESPONSE)
```

predito	0	1
0	103	216
1	13	68

Obtemos assim a seguinte [matriz de confusão](#):

Logo, a nossa taxa de acerto (acurácia) nesse modelo é dada por:

$$ACC = \frac{\text{sum}(\text{diag}(MC))}{\text{sum}(MC)}$$

ACC

```
> ACC
[1] 0.4275
```



# Obrigada!

---

Edmila Montezani  
edmila@gmail.com